УДК 519.6

С. Н. Пупырев

ЭФФЕКТИВНОЕ ВЫЧИСЛЕНИЕ МЕТРИК ЭСТЕТИЧНОСТИ ГРАФА*

1. Введение

Задача визуализации графов состоит в создании простого и интуитивно ясного изображения. На сегодняшний день широко распространено использование автоматических систем для визуализации. Такие системы требуют создания и изучения специализированных алгоритмов укладки графов. Вопрос об оценке качества того или иного алгоритма обычно решается при помощи вычисления метрик эстетичности графа, полученного применением этого алгоритма. Для двух произвольных изображений невозможно в общем случае сказать, что первое лучше второго, или наоборот. Можно говорить лишь о том, что одно из изображений больше соответствует конкретному критерию эстетичности, чем другое. Данная статья посвящена исследованию общепринятых метрик [1], вычисление которых позволяет судить, насколько укладка графа удовлетворяет критерию эстетичности.

Рассматривается пять метрик эстетичности: унификация длин ребер, максимизация разрешения, ортогональность изображения, соответствие расстояний в графе и минимизация пересечений. Для исследования были выбраны именно эти метрики, так как они наиболее часто используются в современных алгоритмах и, по всей видимости, наилучшим образом описывают качество изображения графа. Данные метрики рассматриваются с точки зрения способа их вычисления на практике. Упор делается на эффективность вычисления, простоту реализации и адекватность результатов. Все метрики рассматриваются для укладок графов в трехмерном пространстве.

Необходимость в эффективных способах вычисления метрик эстетичности обусловлена несколькими причинами. Во-первых, редактирование укладок графов в интерактивном режиме в современных графовых редакторах является очень ресурсоемкой операцией. Особенно это критично при работе с графами больших размеров. Удобной возможностью при работе с графами был бы сбор и вычисление статистики в режиме реального времени,

*Работа выполнена при поддержке программы «Развитие научного потенциала высшей школы», проект № 2.1.1/3537.

которая бы позволяла оценивать качество изображения графа. Отсутствие подобной возможности в современных графовых системах не в последнюю очередь обусловлено сложностью реализации алгоритмов вычисления метрик. Во-вторых, метрики эстетичности часто используются при разработке новых алгоритмов укладки, а также иногда являются их составляющими частями. Например, в генетических алгоритмах [2] вычисление метрик – самый ресурсоемкий этап, поэтому он должен быть максимально эффективен.

Необходимые определения даны в разделе 2. Описание самих метрик эстетичности и способов их вычисления приведены в разделе 3. Раздел 4 содержит результаты тестирования предложенных алгоритмов и оценку полученных результатов.

2. Определения

Граф G состоит из множества вершин V и множества ребер E. Вершина v – произвольный элемент множества V, ребро e – пара вершин. Количество вершин в графе обозначается $n = |V|$, количество ребер $m = |E|$. Две вершины в графе называются смежными, если пара этих вершин содержится во множестве ребер. Через $degree(v)$ будем обозначать количество ребер, инцидентных вершине v.

Укладкой графа называют отображение вершин в множество точек и его ребер в жордановы кривые плоскости или пространства. В данной статье будут рассматриваться укладки обыкновенных связных неориентированных графов в трехмерном пространстве. Все изложенные алгоритмы естественным образом обобщаются на более общий случай.

Метрики для пяти эстетических критериев представлены ниже. Каждая метрика – это вещественное число в интервале $[0, 1]$. Будем считать, что укладка полностью соответствует критерию, если метрика равна 1, и полностью не соответствует, если метрика равна 0. Например, плоская укладка (в двумерном случае) будет иметь метрику 1 для критерия «минимизация числа пересечений ребер». Для каждого эстетического критерия предложен алгоритм вычисления его метрики.

3. Метрики эстетичности

3.1. Унификация длин ребер

Этот критерий предполагает минимизацию максимального различия между длинами линий, изображающих ребра графа. Для вычисления этой метрики понадобится средняя длина ребра в графе

$$AvgEdge = \frac{\sum length(e_i)}{m}, \quad e_i \in E,$$

где суммирование ведется по всем ребрам графа. Здесь $length(e_i)$ – длина ребра e_i (длина кривой, изображающей ребро). Общее значение метрики вычисляется по формуле

$$\mathrm{N} = 1 - \frac{\sum |length(e_i) - AvgEdge|}{m}, \quad e_i \in E.$$

При равной длине всех ребер метрика будет иметь значение 1.

3.2. Минимальный угол между двумя смежными ребрами

В данном критерии рассматриваются углы между ребрами, исходящими из одной вершины. Для хорошего изображения необходимо, чтобы ребра расходились как можно шире.

$$\mathrm{N} = \frac{1}{n} \cdot \sum_{i=1}^{n} \frac{|\theta_{v_i} - \theta_{v_{imin}}|}{\theta_{v_i}},$$

где $\theta_{v_i} = 2\pi/degree(v_i)$ – максимально возможный угол при вершине v_i, а $\theta_{v_{imin}}$ – минимальный угол между ребрами, исходящими из этой вершины. Суммирование ведется по всем вершинам, которым инцидентно более одного ребра.

3.3. Ортогональность

Напомним, что изображение графа называется ортогональным, если каждое ребро изображается в виде ломаной линии, состоящей из сегментов, параллельных осям координат. Ортогональные изображения широко используются в различных приложениях (печатные схемы, диаграммы, проектирование баз данных), поэтому данный критерий эстетичности важен для большого числа алгоритмов, строящих ортогональные или близкие к ортогональным укладкам. Определим θ_i – минимальный угол отклонения ребра e_i от оси координат (будем рассматривать минимальный угол отклонения от осей x, y или z). Тогда метрика ортогональности задается следующим образом:

$$\mathrm{N} = 1 - \frac{1}{m} \cdot \sum \frac{\theta_i}{90^\circ}.$$

Если каждое ребро будет параллельно какой-нибудь оси ($\theta_i = 0^\circ$ или $\theta_i = 180^\circ$), то $\mathrm{N} = 1$, т. е. изображение полностью ортогонально.

Все вышеприведенные метрики естественным образом вычисляются за время $O(n+m)$ и при таком же количестве памяти. Более сложный случай со следующими двумя метриками. Эффективного способа их вычисления, насколько нам известно, до сих пор не приводилось.

3.4. Соответствие расстояния между вершинами в графе и в изображении

Считается, что в хорошем изображении расстояние между вершинами должно соответствовать минимальному расстоянию между этими вершинами в графе. Рассматриваемый критерий эстетичности является мерой такого соответствия. Будем считать, что расстояние между вершинами v_1 и v_2 оптимальное, если

$$\frac{dist(P_{v_1}, P_{v_2})}{dist(v_1, v_2)} = AvgEdge,$$

где $dist(P_{v_1}, P_{v_2})$ – расстояние между точками, изображающими вершины v_1 и v_2; $dist(v_1, v_2)$ – расстояние между вершинами v_1 и v_2 в графе. Общая оценка задается следующим образом:

$$\mathrm{N} = 1 - \frac{1}{m} \sum \frac{|dist(P_{v_1}, P_{v_2}) - dist(v_1, v_2) \cdot AvgEdge|}{dist(v_1, v_2) \cdot AvgEdge}, \quad v_1, v_2 \in V. \qquad (1)$$

Для вычисления метрики по данной формуле требуется найти расстояния между всеми парами вершин в графе, т. е. затратить $O(nm)$ времени и $O(n^2)$ памяти, что может быть неприемлемо при работе с большими графами (более тысячи вершин). Предлагается новая техника, основанная на кластеризации графа, позволяющая снизить сложность выполнения алгоритма. Основная идея подхода в том, что качественное изображение графа должно быть качественным как на локальном, так и на глобальном уровне (впервые понятие локального и глобального уровня графа рассмотрено в [3]). Граф разбивается на несколько кластеров (связных подграфов) небольшого размера. Вычисление метрики осуществляется отдельно для каждого кластера (локальный уровень). После этого строится новый граф, вершинами которого являются построенные на предыдущем шаге кластеры. Далее метрика вычисляется для нового графа (глобальный уровень). Итоговая оценка получается комбинацией результатов на локальном и глобальном уровнях. Таким образом, мы уменьшаем количество вершин, для которых работает алгоритм. Ниже приведено описание алгоритма.

Масштабируемый алгоритм

1. Разбить граф G на набор непересекающихся кластеров C_i.

2. Построить новый граф G'. Вершинами G' являются кластеры C_i,

причем между C_i и C_j есть ребро тогда и только тогда, когда существует путь из какой-либо вершины кластера C_i в вершину кластера C_j.

3. Вычислить метрики эстетичности для каждого из кластеров C_j и для графа G'.

4. Итоговая метрика получается комбинацией результатов предыдущего шага.

На первом шаге алгоритма требуется разбить граф на непересекающиеся подграфы (кластеры). Кластеры следует выбирать таким образом, чтобы граф, составленный из этих кластеров был максимально похож на исходный. Новый граф G' – это уменьшенная копия графа G, имеющая такую же структуру, но без мелких деталей. Предлагается следующая схема выбора кластеров. На каждом шаге выбираем произвольную непомеченную вершину и в качестве кластера берем подграф, содержащий эту вершину и все непомеченные вершины, лежащие от нее на некотором расстоянии R. После этого помечаем все вершины, попавшие в кластер. Так действуем, пока в графе еще остались непомеченные вершины. Константу R нужно подобрать, исходя из структуры исходного графа, таким образом, чтобы в кластер попадало не более $\sqrt{n}$ вершин. Если граф довольно плотный, то есть вероятность, что в кластер попадет большее количество вершин. В этом случае необходимо ограничить размер кластера. На втором шаге алгоритма строится граф кластеров G'. Как отмечалось выше, вершинами этого графа являются кластеры. Между двумя кластерами C_i и C_j есть ребро, если существует путь из вершин первого кластера во второй (в случае связного графа G такой путь будет существовать всегда). Длина этого ребра – среднее расстояние между всеми вершинами кластера C_i и кластера C_j. Далее вычисляются метрики эстетичности набора графов по формуле (1). В зависимости от области применения алгоритма локальные и глобальные метрики могут использоваться различными способами. Общая метрика может быть вычислена, например, как линейная комбинация локальной и глобальной метрики.

Оценим временную сложность алгоритма. Пусть максимальный размер построенного кластера равен k. Тогда всего будет не более $\frac{n}{k}$ кластеров и общая сложность выполнения составит $O\left(\left(\frac{n}{k}\right)^3 + \frac{n}{k} \cdot k^3\right)$, где первое слагаемое соответствует нахождению кратчайших расстояний между всеми парами вершин в графе G', а второе – выполнению аналогичной операции для каждого кластера C_i. Отсюда видно, что для минимизации времени выполнения требуется выбирать размер кластера равным $k = \sqrt[5]{\frac{3}{2}n^2} \approx \sqrt{n}$. В этом случае общая оценка будет следующей: $O\left(\left(\frac{n}{k}\right)^3 + \frac{n}{k} \cdot k^3\right) = O\left(n \cdot \frac{n}{k} + n \cdot n\right) = O(n^2)$.

3.5. Пересечение ребер

Пересечение ребер для укладки графа на плоскости – один из наиболее важных критериев эстетичности. Однако при его оценке в трехмерном пространстве возникает ряд трудностей. Главная проблема в том, что почти любая укладка графа не будет содержать пересечений ребер в пространстве. Поступим следующим образом. Спроецируем все вершины и ребра графа на несколько случайно выбранных плоскостей и посчитаем количество пересечений ребер в получившихся двумерных укладках. Среднее количество пересечений C_{avg} является адекватной мерой количества пересечений ребер для трехмерного изображения. Это легко объяснимо: большинство современных средств просмотра графики и изображений позволяют просматривать только проекции трехмерной укладки графа. Соответственно качество изображения оценивается именно по таким проекциям. Взяв несколько «независимых» проекций (в таком качестве вполне подойдет случайный набор), получим хорошее приближение к реальной оценке.

Для корректного определения рассматриваемой метрики необходимо знать максимально количество пересечений ребер для всех возможных укладок данного графа. На сегодняшний момент точная верхняя оценка этого числа неизвестна [1]. Тем не менее резонно предположить, что «наихудший» случай достигается для случайной укладки. Под случайной здесь понимается укладка, в которой вероятность расположения любой вершины графа одинакова для всех точек плоскости. Используя данное свойство, можно получить хорошее приближение к максимальному числу пересечений ребер. Для этого посчитаем количество пересечений ребер для некоторого небольшого числа случайных укладок графа и среди них выберем максимум. Обозначим полученное число C_{mx}. Общее значение метрики вычисляется по формуле

$$\mathrm{N} = 1 - \begin{cases} \frac{C_{avg}}{C_{mx}}, & \text{если } C_{mx} > 0, \\ 0, & \text{если } C_{mx} = 0, \end{cases}$$

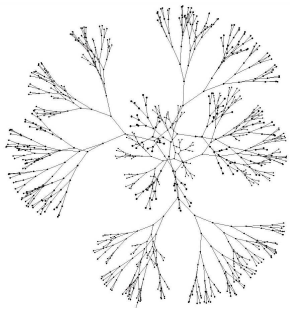
где C – число пересечений ребер в рассматриваемой проекции изображения графа на плоскость.

Для подсчета этой метрики требуется вычислять количество пересечений ребер графа на плоскости. Эта задача очевидным образом сводится к известной задаче о пересечении отрезков на плоскости, алгоритм решения которой описан в [4]. Этот алгоритм имеет сложность $O(P + n \log n)$, где P – количество пересечений. В прикладных задачах чаще всего большие графы сильно разряжены, т. е. справедливо равенство $m = O(n)$. В этом случае оценка принимает вид $O(P + n \log n) = O(n^2 + n \log n) = O(n^2)$.

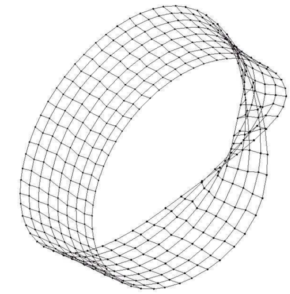
4. Примеры вычисления метрик

Данный раздел содержит демонстрацию использования описанных метрик эстетичности. Все алгоритмы реализованы в системе визуализации графов GraphVis [5], созданной автором. Тестирование проводилось на машине Pentium IV 1,8 GHz с 500 Mb оперативной памяти. Для построения укладок графов использовался директивный метод, описанный в [6].

Для тестирования алгоритма было выбрано несколько графов из известного пакета графов AT&T graph library [7] (рис. 2–3). Также рассматриваются два графа (см. рис. 1), которые, по мнению автора, хорошо иллюстрируют работу алгоритмов. Рис. 1, *а* содержит полное бинарное дерево с 1023 вершинами, рис. 1, *б* – замкнутую с двух сторон сетку (лист Мебиуса).

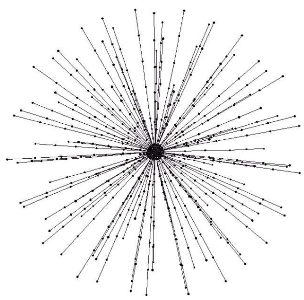


а

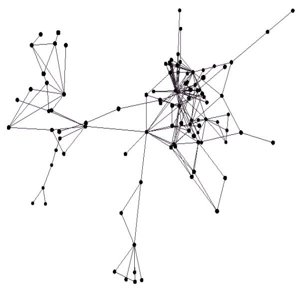


б

Рис. 1



а



б

Рис. 2

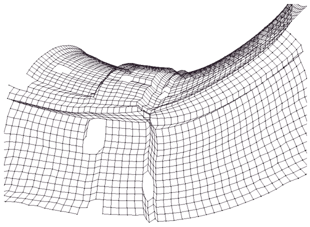


Рис. 3

Сравнение результатов работы предлагаемого алгоритма для различных графов

Номера рисунков	Размер графа, вершин/ребер	Время выполнения алгоритма, с	Унификация длин	Угол между ребрами	Ортогональность	Соответствие расстояний	Пересечение ребер
1,*а*	1023/1022	0.1	0.83	0.62	0.40	0.48	0.97
1,*б*	450/7650	0.2	0.81	0.83	0.31	0.78	0.99
2,*а*	601/600	0.1	0.4	0.99	0.4	0.58	0.89
2,*б*	100/131	0.1	0.8	0.51	0.36	0.64	0.87
3	2576/5019	0.3	0.92	0.74	0.81	0.60	0.96

Результаты тестирования приведены в таблице. Значение метрики *Пересечение ребер* близко к оптимуму для всех укладок. Это показатель того, что получены качественные изображения графов, подчеркивающие их структуру. Все изображения являются «почти планарными», т. е. содержат малое количество пересечений ребер. Используемый алгоритм [6] строит укладки графов, в которых расстояние между вершинами равно расстоянию между этими вершинами в графе. Это объясняет близкие к 1 значения метрик *Унификация длин ребер* и *Соответствие расстояний*. Практически все укладки, полученные директивным методом, слабо ортогональные, т. е. не соответствуют метрике *Ортогональность* более чем на 50 %. Можно сказать, что методы, основанные на физических аналогиях, не учитывают этот критерий эстетичности.

Все метрики были вычислены за время, не превосходящее 0.3 с. Дополнительная память при вычислениях, помимо памяти для хранения самого графа, не использовалась. Поэтому можно утверждать, что разработанные способы вычисления метрик эффективно работают с графами больших размеров (до 10 000 вершин). Кроме того, полученные значения метрик правдоподобно отражают структурные особенности укладки графов. Таким образом, описанные алгоритмы могут применяться для вычисления метрик эстетичности и оценки качества изображения графа.

1. Purchase H. Graph drawing aesthetics // J. Visual Languages and Computing. 2002. Vol. 13, № 5. P. 501–516.
2. Rosete A., Ochoa A. Genetic graph drawing // Proc. of the 13th Int. Conf. of Applications of Artificial Intelligence in Engineering. 1998. P. 37–41.
3. Harel D., Koren Y. A fast multi-scale method for drawing large graphs // Graph Drawing. [Lecture Notes in Computer Science; Vol. 1984]. B.; Heidelberg: Springer, 2001. P. 183–196.
4. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. М.: МЦНМО, 2001.
5. Система визуализации графов GraphVis. `http://code.google.com/p/graphvis`
6. Kamada T., Kawai S. An algorithm for drawing general undirected graphs // Information Processing Lett. 1989. Vol. 31, № 1. P. 7–15.
7. The AT&T graph collection. `http://www.graphdrawing.org`

Статья поступила 20.09.2007, окончательный вариант 01.11.2007